

**WEST**

Generate Collection

L3: Entry 1 of 3

File: USPT

May 16, 2000

US-PAT-NO: 6065019

DOCUMENT-IDENTIFIER: US 6065019 A

TITLE: Method and apparatus for allocating and freeing storage utilizing multiple tiers of storage organization

DATE-ISSUED: May 16, 2000

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ault; Donald F.	Hyde Park	NY		
Fischer; John F.	Wappingers Falls	NY		
Miller; Eric T.	Wappingers Falls	NY		

US-CL-CURRENT: 707/206; 711/171

## CLAIMS:

What is claimed is:

1. In an information handling system having a predetermined routine for obtaining storage from a repository, said predetermined routine requiring a lock for serialization, a method for obtaining storage for a requester while minimizing use of said predetermined routine, comprising the steps of:  
maintaining a set of previously assigned storage blocks that have been released by requesters;  
maintaining a set of unassigned storage blocks that have been obtained from said repository using said predetermined routine but have not been assigned to a requester;  
completing an incoming storage request from said previously assigned storage blocks without acquiring a lock for serialization if said storage request can be completed from said previously assigned storage blocks;  
completing said storage request from said unassigned storage blocks without acquiring a lock for serialization if said storage request cannot be completed from said previously assigned storage blocks but can be completed from said unassigned storage blocks; and  
completing said request from said repository using said predetermined routine if said storage request cannot be completed from either said previously assigned storage blocks or said unassigned storage blocks.
2. The method of claim 1 wherein said first maintaining step comprises the step of:  
maintaining a plurality of sets of previously assigned storage blocks, the storage blocks in each of said sets having a predetermined size.
3. The method of claim 1 wherein said second maintaining step comprises the step of:  
maintaining a plurality of sets of unassigned storage blocks, the storage blocks in each of said sets having a predetermined size.
4. The method of claim 1 wherein said step of completing an incoming storage request from said previously assigned storage blocks is performed using an atomic operation.
5. The method of claim 1 wherein said step of completing an incoming storage request from said unassigned storage blocks is performed using an atomic operation.
6. The method of claim 1 in which said set of previously assigned storage blocks is initially empty.
7. The method of claim 1 in which said set of unassigned storage blocks is

initially empty.

8. The method of claim 1 in which said previously assigned storage blocks are maintained as a linked list.

9. The method of claim 1 in which said previously assigned storage blocks are maintained as a LIFO data structure.

10. The method of claim 1, further comprising the steps of:  
maintaining counts of the number of blocks requested for each of a plurality of storage sizes; and  
determining one or more optimum block sizes based upon said counts.

11. The method of claim 1 wherein said step of completing an incoming storage request from said previously assigned storage blocks is performed using an atomic instruction in which a first value is compared with a second value to produce a comparison result and, depending on the comparison result, replaced with a third value.

12. The method of claim 1 wherein said step of completing said storage request from said unassigned storage blocks is performed using an atomic instruction in which a first value is compared with a second value to produce a comparison result and, depending on the comparison result, replaced with a third value.

13. In an information handling system having a predetermined routine for obtaining storage from a repository, said predetermined routine requiring a lock for serialization, apparatus for obtaining storage for a requester while minimizing use of said predetermined routine, comprising:

means for maintaining a set of previously assigned storage blocks that have been released by requesters;

means for maintaining a set of unassigned storage blocks that have been obtained from said repository using said predetermined routine but have not been assigned to a requester;

means for completing an incoming storage request from said previously assigned storage blocks without acquiring a lock for serialization if said storage request can be completed from said previously assigned storage blocks;

means for completing said storage request from said unassigned storage blocks without acquiring a lock for serialization if said storage request cannot be completed from said previously assigned storage blocks but can be completed from said unassigned storage blocks; and

means for completing said request from said repository using said predetermined routine if said storage request cannot be completed from either said previously assigned storage blocks or said unassigned storage blocks.

14. The apparatus of claim 13 wherein said first maintaining means comprises:  
means for maintaining a plurality of sets of previously assigned storage blocks, the storage blocks in each of said sets having a predetermined size.

15. The apparatus of claim 13 wherein said second maintaining means comprises:  
means for maintaining a plurality of sets of unassigned storage blocks, the storage blocks in each of said sets having a predetermined size.

16. The apparatus of claim 13, further comprising:  
means for maintaining counts of the number of blocks requested for each of a plurality of storage sizes; and

means for determining one or more optimum block sizes based upon said counts.

17. The apparatus of claim 13 wherein said means for completing an incoming storage request from said previously assigned storage blocks uses an atomic instruction in which a first value is compared with a second value to produce a comparison result and, depending on the comparison result, replaced with a third value.

18. The apparatus of claim 13 wherein said means for completing said storage request from said unassigned storage blocks uses an atomic instruction in which a first value is compared with a second value to produce a comparison result and, depending on the comparison result, replaced with a third value.

19. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for obtaining storage for a requester while minimizing use of a predetermined routine for obtaining storage from a repository in an information handling system having said routine, said predetermined routine requiring a lock for serialization said method steps comprising:

maintaining a set of previously assigned storage blocks that have been released by requesters;

maintaining a set of unassigned storage blocks that have been obtained from said repository using said predetermined routine but have not been assigned to a requester;

completing an incoming storage request from said previously assigned storage blocks without acquiring a lock for serialization if said storage request can be

completed from said previously assigned storage blocks;  
completing said storage request from said unassigned storage blocks without acquiring a lock for serialization if said storage request cannot be completed from said previously assigned storage blocks but can be completed from said unassigned storage blocks; and  
completing said request from said repository using said predetermined routine if said storage request cannot be completed from either said previously assigned storage blocks or said unassigned storage blocks.

20. The program storage device of claim 19 wherein said first maintaining step comprises the step of:

maintaining a plurality of sets of previously assigned storage blocks, the storage blocks in each of said sets having a predetermined size.

21. The program storage device of claim 19 wherein said second maintaining step comprises the step of:

maintaining a plurality of sets of unassigned storage blocks, the storage blocks in each of said sets having a predetermined size.

22. The program storage device of claim 19 wherein said step of completing an incoming storage request from said previously assigned storage blocks is performed using an atomic operation.

23. The program storage device of claim 19 wherein said step of completing an incoming storage request from said unassigned storage blocks is performed using an atomic operation.

24. The program storage device of claim 19 in which said set of previously assigned storage blocks is initially empty.

25. The program storage device of claim 19 in which said set of unassigned storage blocks is initially empty.

26. The program storage device of claim 19 in which said previously assigned storage blocks are maintained as a linked list.

27. The program storage device of claim 19 in which said previously assigned storage blocks are maintained as a LIFO data structure.

28. The program storage device of claim 19, said method steps further comprising: maintaining counts of the number of blocks requested for each of a plurality of storage sizes; and

determining one or more optimum block sizes based upon said counts.

29. The program storage device of claim 19 wherein said step of completing an incoming storage request from said previously assigned storage blocks is performed using an atomic instruction in which a first value is compared with a second value to produce a comparison result and, depending on the comparison result, replaced with a third value.

30. The program storage device of claim 19 wherein said step of completing said storage request from said unassigned storage blocks is performed using an atomic instruction in which a first value is compared with a second value to produce a comparison result and, depending on the comparison result, replaced with a third value.